503Project Models LDA, GLMN, and NSC

Maha Jayapal

6/8/2022

Loading the packages

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(e1071)
library(Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:e1071':
##
##     impute
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(plyr)
```

```
##
## Attaching package: 'plyr'

## The following objects are masked from 'package:Hmisc':
##
##     is.discrete, summarize
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
stroke<- read.csv('c:\\maha\\503\\healthcare-dataset-stroke-data.csv', header = TRUE)
```

```
str(stroke)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
stroke$gender <- as.numeric(as.factor(stroke$gender))
stroke$ever_married <- as.numeric(as.factor(stroke$ever_married))
stroke$work_type <- as.numeric(as.factor(stroke$work_type))
stroke$Residence_type <- as.numeric(as.factor(stroke$Residence_type))
stroke$smoking_status <- as.numeric(as.factor(stroke$smoking_status))
stroke$bmi <- as.numeric(stroke$bmi)
```

```
## Warning: NAs introduced by coercion
```

```
levels(stroke$stroke) <- c("YES", "NO")
#stroke$stroke <- as.factor(stroke$stroke)
```

```
stroke$stroke<-ifelse(stroke$stroke == 1,"YES","NO")
table(stroke$stroke)
```

```
##
##   NO  YES
## 4861  249
```

```
trainingRows <- createDataPartition(stroke$stroke, p = .80, list = FALSE)
stroke_train <- stroke[trainingRows, ]
stroke_test <- stroke[-trainingRows, ]
stroke_trainx <- stroke_train[,1:11]
stroke_trainy <- as.factor(stroke_train[, 12])
stroke_trainimp <- preProcess(stroke_trainx, "knnImpute")
stroke_trainxpr <- predict(stroke_trainimp, stroke_trainx)
stroke_testxpr <- predict(stroke_trainimp, stroke_test)
```

```
ctrl <- trainControl(method = "cv",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     savePredictions = TRUE)
```

```
set.seed(500)
ldaFit_stroke <- train(x = stroke_trainxpr,
                       y = stroke_trainy,
                       method = "lda",
                       preProc = c("center","scale"),
                       metric = "ROC",
                       trControl = ctrl)
ldaFit_stroke
```

```
## Linear Discriminant Analysis
##
## 4089 samples
##   11 predictor
##    2 classes: 'NO', 'YES'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3680, 3681, 3680, 3680, 3680, 3680, ...
## Resampling results:
##
##   ROC       Sens       Spec
##   0.834545  0.9894575  0.075
```

```
glmnGrid <- expand.grid(alpha = c(0, .1, .2, .4, .6, .8, 1),
lambda = seq(.01, .2, length = 10))
set.seed(500)
glmnFit_stroke <- train(x = stroke_trainxpr,
                        y = stroke_trainy,
                        method = "glmnet",
                        tuneGrid = glmnGrid,
```

```
                    preProc = c("center", "scale"),
                    metric = "ROC",
                    trControl = ctrl)
glmnFit_stroke
```

```
## glmnet
##
## 4089 samples
##   11 predictor
##    2 classes: 'NO', 'YES'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3680, 3681, 3680, 3680, 3680, 3680, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda      ROC        Sens  Spec
##   0.0    0.01000000  0.8408024  1     0
##   0.0    0.03111111  0.8345654  1     0
##   0.0    0.05222222  0.8300901  1     0
##   0.0    0.07333333  0.8270042  1     0
##   0.0    0.09444444  0.8248056  1     0
##   0.0    0.11555556  0.8231984  1     0
##   0.0    0.13666667  0.8218741  1     0
##   0.0    0.15777778  0.8209869  1     0
##   0.0    0.17888889  0.8202668  1     0
##   0.0    0.20000000  0.8195338  1     0
##   0.1    0.01000000  0.8426024  1     0
##   0.1    0.03111111  0.8389368  1     0
##   0.1    0.05222222  0.8366223  1     0
##   0.1    0.07333333  0.8357351  1     0
##   0.1    0.09444444  0.8358509  1     0
##   0.1    0.11555556  0.8361981  1     0
##   0.1    0.13666667  0.8374457  1     0
##   0.1    0.15777778  0.8386930  1     0
##   0.1    0.17888889  0.8395029  1     0
##   0.1    0.20000000  0.8406866  1     0
##   0.2    0.01000000  0.8444798  1     0
##   0.2    0.03111111  0.8431155  1     0
##   0.2    0.05222222  0.8418174  1     0
##   0.2    0.07333333  0.8417020  1     0
##   0.2    0.09444444  0.8422813  1     0
##   0.2    0.11555556  0.8392610  1     0
##   0.2    0.13666667  0.8373777  1     0
##   0.2    0.15777778  0.8402440  1     0
##   0.2    0.17888889  0.8402440  1     0
##   0.2    0.20000000  0.8402440  1     0
##   0.4    0.01000000  0.8450067  1     0
##   0.4    0.03111111  0.8441712  1     0
##   0.4    0.05222222  0.8389008  1     0
##   0.4    0.07333333  0.8402183  1     0
##   0.4    0.09444444  0.8402440  1     0
##   0.4    0.11555556  0.8402440  1     0
```

```
##   0.4    0.13666667  0.6205720  1     0
##   0.4    0.15777778  0.5000000  1     0
##   0.4    0.17888889  0.5000000  1     0
##   0.4    0.20000000  0.5000000  1     0
##   0.6    0.01000000  0.8452384  1     0
##   0.6    0.03111111  0.8392990  1     0
##   0.6    0.05222222  0.8402440  1     0
##   0.6    0.07333333  0.8402440  1     0
##   0.6    0.09444444  0.5000000  1     0
##   0.6    0.11555556  0.5000000  1     0
##   0.6    0.13666667  0.5000000  1     0
##   0.6    0.15777778  0.5000000  1     0
##   0.6    0.17888889  0.5000000  1     0
##   0.6    0.20000000  0.5000000  1     0
##   0.8    0.01000000  0.8451744  1     0
##   0.8    0.03111111  0.8402183  1     0
##   0.8    0.05222222  0.8402440  1     0
##   0.8    0.07333333  0.5000000  1     0
##   0.8    0.09444444  0.5000000  1     0
##   0.8    0.11555556  0.5000000  1     0
##   0.8    0.13666667  0.5000000  1     0
##   0.8    0.15777778  0.5000000  1     0
##   0.8    0.17888889  0.5000000  1     0
##   0.8    0.20000000  0.5000000  1     0
##   1.0    0.01000000  0.8435415  1     0
##   1.0    0.03111111  0.8402440  1     0
##   1.0    0.05222222  0.8402440  1     0
##   1.0    0.07333333  0.5000000  1     0
##   1.0    0.09444444  0.5000000  1     0
##   1.0    0.11555556  0.5000000  1     0
##   1.0    0.13666667  0.5000000  1     0
##   1.0    0.15777778  0.5000000  1     0
##   1.0    0.17888889  0.5000000  1     0
##   1.0    0.20000000  0.5000000  1     0
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 0.6 and lambda = 0.01.
```

```
set.seed(500)
nscFit_stroke <- train(x = stroke_trainxpr,
                       y = stroke_trainy,
                       method = "pam",
                       preProc = c("center", "scale"),
                       tuneGrid = data.frame(threshold = seq(0, 25, length = 30)),
                       metric = "ROC",
                       trControl = ctrl)
```

```
## 11111111111
```

```
nscFit_stroke
```

```
## Nearest Shrunken Centroids
##
```

```
## 4089 samples
##   11 predictor
##    2 classes: 'NO', 'YES'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 3680, 3681, 3680, 3680, 3680, 3680, ...
## Resampling results across tuning parameters:
##
##   threshold  ROC        Sens  Spec
##    0.000000  0.8142112  1     0
##    0.862069  0.8216290  1     0
##    1.724138  0.8287395  1     0
##    2.586207  0.8358759  1     0
##    3.448276  0.8410218  1     0
##    4.310345  0.8384380  1     0
##    5.172414  0.8402440  1     0
##    6.034483  0.8402440  1     0
##    6.896552  0.8402440  1     0
##    7.758621  0.6205720  1     0
##    8.620690  0.5000000  1     0
##    9.482759  0.5000000  1     0
##   10.344828  0.5000000  1     0
##   11.206897  0.5000000  1     0
##   12.068966  0.5000000  1     0
##   12.931034  0.5000000  1     0
##   13.793103  0.5000000  1     0
##   14.655172  0.5000000  1     0
##   15.517241  0.5000000  1     0
##   16.379310  0.5000000  1     0
##   17.241379  0.5000000  1     0
##   18.103448  0.5000000  1     0
##   18.965517  0.5000000  1     0
##   19.827586  0.5000000  1     0
##   20.689655  0.5000000  1     0
##   21.551724  0.5000000  1     0
##   22.413793  0.5000000  1     0
##   23.275862  0.5000000  1     0
##   24.137931  0.5000000  1     0
##   25.000000  0.5000000  1     0
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was threshold = 3.448276.
```